

Help Topic: Poking the MOOSDB

Spring 2020

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

Poking the MOOSDB

Poking refers to the idea of publishing a variable-value pair to the **MOOSDB**. Many apps publish to the **MOOSDB** during the course of normal operation. Poking implies a publication that perhaps was not planned, or outside the normal mode of business. It is often very useful for debugging. Here we describe the **uPokeDB** tool.

Where to get more information:

- **uPokeDB**: <http://oceanai.mit.edu/ivpman/apps/uPokeDB>

Or simply:

```
$ uPokeDB --web
```

Poking the MOOSDB with uPokeDB

uPokeDB is a command-line tool for poking the **MOOSDB** with one or more variable-value pairs. Poking the **MOOSDB** requires knowing where the **MOOSDB** is running in terms of its IP address, (**ServerHost** parameter), and port number, (**ServerPort** parameter). These may be specified on the command line to **uPokeDB**, but for our purposes here we assume the existence of a mission file, `alpha.moos` with this information:

```
// (wget http://oceanai.mit.edu/2.680/examples/alpha.moos)
ServerHost = localhost
ServerPort = 9000
Community = alpha
```

Your goals in this part are:

1. Open two terminal windows and launch the **MOOSDB** and **uXMS** as done previously:

```
$ MOOSDB alpha.moos
$ uXMS alpha.moos --all
```

Now open a third terminal window for poking the **MOOSDB** as follows:

```
$ uPokeDB DEPLOY=true SPEED=2 alpha.moos
```

Note the two new variables, `DEPLOY` and `SPEED`, appearing in the `uXMS` window. It should look something like:

```
=====
uXMS_655                                     0/0(204)
=====
VarName      (S) (T) (C)  VarValue (SCOPING:EVENTS)
-----
DB_CLIENTS           "uXMS_655,"
DB_TIME             1386249435.276804
DB_UPTIME           46.213629
DEPLOY              "true"
SPEED               2
```

- Note the variable values in `uXMS`. `DEPLOY` has the value `"true"` with double quotes, indicating that it is a string. The variable `SPEED` is of type double, indicated by the lack of quotes. The types were inferred by `uPokeDB` by heuristically checking whether the arguments are numerical or not. But sometimes you *do* want to publish a string with a numerical value. Try posting the variable `HEIGHT` with the string value of `"192"`, noting the colon-equals instead of equals:

```
$ uPokeDB HEIGHT:=192 alpha.moos
```

Note the new variable, `HEIGHT`, appearing in the `uXMS` window. It should look something like the below output, where in this case, the (S)ource column is expanded to show the source of the postings.

```
=====
uXMS_655                                     0/0(347)
=====
VarName      (S)ource      (T) (C)  VarValue (SCOPING:EVENTS)
-----
DB_CLIENTS   MOOSDB_alpha           "uXMS_655,"
DB_TIME      MOOSDB_alpha           1386250092.847527
DB_UPTIME    MOOSDB_alpha           703.784353
DEPLOY       uPokeDB                 "true"
HEIGHT       uPokeDB                 "192"
SPEED        uPokeDB                 2
```

Further things to try Using `uPokeDB`

Here's some other things to consider and try:

- Trying poking the `DEPLOY` variable to the `MOOSDB` a second time, this time with:

```
$ uPokeDB DEPLOY=100 alpha.moos
```

Does the value of `DEPLOY` change? If not, why not?

- Create a simple script of pokes on the command line as follows:

```
$ uPokeDB APPLES=1 alpha.moos; sleep 5; uPokeDB APPLES=2 alpha.moos;
```

If you're new to the command line environment, the semicolon above separates successful command line invocations. The `sleep` command is a common shell utility that will simply pause a given number of seconds before completing.

3. Another way to execute the same simple script as above is to store the above three commands in a file named, for example, `myscript`:

```
uPokeDB APPLES=1 alpha.moos  
sleep 5  
uPokeDB APPLES=2 alpha.moos
```

With the above file you can make the two successive pokes to the `MOOSDB`, with five seconds in between, with:

```
$ source myscript
```

There are many other ways of poking the `MOOSDB`. All MOOS apps that publish anything are examples. Of course many MOOS applications publish a fixed set of variables that are not easily changeable without re-coding. But certain apps like `uTimerScript` and `pMarineViewer` have built-in configuration file parameters for poking the `MOOSDB` in user configurable ways.